PROGRAM 2:

1) Develop a lexical Analyzer to identify whether a given line is a comment or not using C  
  
CODE: #include <stdio.h>

#include <string.h>

void checkComment(char \*line) {

// Check if the line starts with a single-line comment

if (line[0] == '/' && line[1] == '/') {

printf("The line is a single-line comment.\n");

}

// Check if the line starts with a multi-line comment

else if (line[0] == '/' && line[1] == '\*') {

int len = strlen(line);

if (len >= 4 && line[len - 2] == '\*' && line[len - 1] == '/') {

printf("The line is a multi-line comment.\n");

} else {

printf("The line starts a multi-line comment but does not end it.\n");

}

}

// If the line is not a comment

else {

printf("The line is not a comment.\n");

}

}

int main() {

// Hardcoded input lines

char line1[] = "// This is a single-line comment";

char line2[] = "/\* This is a multi-line comment \*/";

char line3[] = "int a = 10;";

char line4[] = "/\* This starts a multi-line comment";

printf("Checking Line 1: %s\n", line1);

checkComment(line1);

printf("\nChecking Line 2: %s\n", line2);

checkComment(line2);

printf("\nChecking Line 3: %s\n", line3);

checkComment(line3);

printf("\nChecking Line 4: %s\n", line4);

checkComment(line4);

return 0;

}

OUTPUT:

Checking Line 1: // This is a single-line comment

The line is a single-line comment.

Checking Line 2: /\* This is a multi-line comment \*/

The line is a multi-line comment.

Checking Line 3: int a = 10;

The line is not a comment.

Checking Line 4: /\* This starts a multi-line comment

The line starts a multi-line comment but does not end it.